**Bài tập**

Cho 2 số nguyên a, b được nhập từ bàn phím. Bạn hãy viết chương trình hiển thị ra màn hình (a OR b) + (a AND b) với OR và AND là 2 phép toán trên bit.

Ví dụ nếu bạn nhập a = 113, b = 93 như bên dưới:

113 93

Thì màn hình sẽ hiển thị ra:

206

**Lý thuyết**

Bản chất dữ liệu sẽ được máy tính lưu dưới dạng dãy các bit (dãy các con số 0 hoặc 1). Ví dụ:

char a = 4;

Biến a sẽ được máy tính lưu trữ bằng 8 bit (kích thước kiểu char là 1 byte mà 1 byte = 8 bit) giống như sau:



Ví dụ khác:

int a = 126;

Biến a sẽ được máy tính lưu trữ bằng 32 bit giống như sau:



Dãy bit trên là biểu diễn của biến a dưới dạng hệ nhị phân, bạn có thể tìm hiểu thêm về cách chuyển đổi sang hệ nhị phân [tại đây](https://codelearn.io/sharing/cach-chuyen-doi-cac-he-so-dem).

Việc biết về các phép toán trên bit có thể giúp bạn viết chương trình chạy nhanh hơn. Ví dụ để kiểm tra một số có phải là số chẵn không bạn có thể làm như sau:

#include<iostream>

using namespace std;

int main() {

int a = 5;

if (a & 1) {

cout << a << " is odd number";

} else {

cout << a << " is even number";

}

return 0;

}

Câu lệnh if (a & 1) sẽ có tốc độ thực thi nhanh hơn if (a % 2 != 0) do đây là phép toán trên bit.

**Các phép toán trên bit:**

|  |  |
| --- | --- |
| Phép AND | & |
| Phép OR | | |
| Phép phủ định NOT | ~ |
| Phép XOR | ^ |
| Phép dịch trái | << |
| Phép dịch phải | >> |

Phép AND giữa 2 bit sẽ trả về 1 khi cả 2 bit đều có giá trị bằng 1, ngược lại nó sẽ trả về 0. Ví dụ:

0 & 0 = 0

0 & 1 = 0

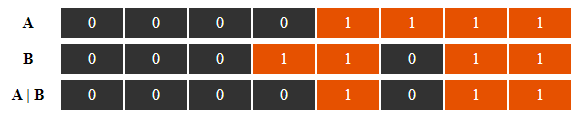
1 & 0 = 0

1 & 1 = 1

Ví dụ về phép AND bit cho 2 số A, B:

A = 15.

B = 27.



Có thể thấy kết quả của A & B là 00001011 (00001011 có giá trị bằng 11 ở hệ cơ số 10). Thực hành phép AND bit trên C++:

#include<iostream>

using namespace std;

int main() {

int a = 15;

int b = 27;

cout << (a & b);

return 0;

}

Kết quả khi chạy chương trình:

11

Phép OR giữa 2 bit sẽ trả về 1 nếu có ít nhất 1 trong 2 bit có giá trị bằng 1, ngược lại nó sẽ trả về 0. Ví dụ:

0 | 0 = 0

0 | 1 = 1

1 | 0 = 1

1 | 1 = 1

Phép NOT là phép đảo bit. Ví dụ:

~ 0 = 1

~ 1 = 0

Phép XOR giữa 2 bit sẽ trả về 1 khi 2 bit có giá trị khác nhau, ngược lại nó sẽ trả về 0. Ví dụ:

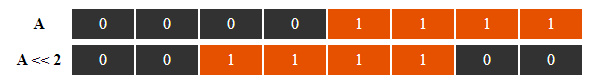
0 ^ 0 = 0

0 ^ 1 = 1

1 ^ 0 = 1

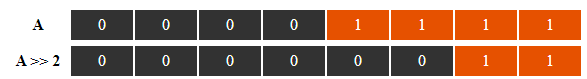
1 ^ 1 = 0

Phép dịch trái là phép dịch tất cả các bit sang trái. Ví dụ:



A << 2 tương đương với tất cả các bit sang trái 2 đơn vị.

Phép dịch phải là phép dịch tất cả các bit sang phải. Ví dụ:



Đọc tới đây bạn đã hiểu về các phép toán trên bit, hãy xem chương trình C++ sau để hiểu rõ hơn:

#include<iostream>

using namespace std;

int main() {

int a = 15;

int b = 27;

cout << "a AND b = " << (a & b) << endl;

cout << "a OR b = " << (a | b) << endl;

cout << "a XOR b = " << (a ^ b) << endl;

cout << "~a = " << (~a) << endl;

cout << "a << 2 = " << (a << 2) << endl;

cout << "a >> 2 = " << (a >> 2) << endl;

return 0;

}

Kết quả khi chạy chương trình:

a AND b = 11

a OR b = 31

a XOR b = 20

~a = -16

a << 2 = 60

a >> 2 = 3

Đọc tới đây bạn đã có thể làm được bài này, hãy quay lại phần bài tập và làm thử.

Nếu bạn chưa làm được thì có thể xem hướng dẫn bên dưới:

**Hướng dẫn**

Code mẫu:

#include<iostream>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

cout << (a | b) + (a & b);

return 0;

}